The moving mesh semi-Lagrangian **MMSISL** method.

S.P. Cook[a,*], C.J. Budd[a], T. Melvin[b]

[a] *University of Bath, BA2 7AY, UK*
[b] *Met Office, FitzRoy Road, Exeter, EX1 3PB, UK*

Abstract

We introduce a novel location-based moving mesh algorithm **MMSISL** in which the arrival points in the Semi-Implicit Semi-Lagrangian (SISL) algorithm are located by using an equidistribution strategy. This algorithm gives a natural coupling between moving mesh methods and SISL methods. It involves little extra cost in implementation as it exploits the interpolation methods already embedded in the SISL algorithm. We apply this method to a number of partial differential equation problems in one-dimension, each of which have sharply defined features. We show that using MMSISL leads to a markedly improved performance over fixed mesh methods, with significantly reduced errors. We also show that unlike many adaptive schemes, no issues arise in the MMSISL algorithm from a CFL condition imposed restriction on the time step.

Keywords: semi-implicit semi-Lagrangian, Burgers' equation, error estimates, moving mesh methods

1. Introduction

1.1. Overview

When using general numerical methods for advection dominated problems, two issues are often encountered in the search for accurate and stable numerical schemes. Firstly, if a small spatial step Δx is used then the time step Δt has also to be small, and is often constrained by the CFL condition $v\Delta t/\Delta x \leq 1$ where v is the advective speed [MM94]. Secondly, many nonlinear advection equations, for example the nonlinear Burgers' equation studied as one of the examples in this paper, develop fronts with small length scales and steep gradients as time evolves. Such fronts need a fine spatial mesh to be resolved accurately. This can be achieved, with significant computational cost, by the use of a globally fine mesh (in both space and time). However, the use of such a fine mesh may not be feasible due to time and cost constraints. Alternatively a resolution of the front can often be achieved, at a smaller computational cost, by using some

*Corresponding author

form of local mesh refinement. However, the use of such a fine spatial mesh, either in the context of a fixed mesh method, or a locally adaptive method, can lead to problems with small time steps as identified above, where the time step is determined from the smallest spatial step in the mesh through the CFL condition. This can lead to a very expensive computation, where the cost of computation in the regions where the mesh is coarse is determined by the time step dictated from the regions where the mesh is fine. This constraint is a severe limitation on the use of fine/locally refined meshes in the context of advection dominated problems. This difficulty is often cited as an argument against the use of adaptive methods in this context [BHR09].

A popular and effective way to overcome many of the problems with the CFL constraint, and which permits the use of a much larger time step, is the SISL algorithm [SC91, Rob81, Rob82, CGM$^+$98, RTS$^+$95, WSW$^+$14]. In this method (to be described in detail in Section 2) the solution is discretised over each time step Δt along a locally defined Lagrangian particle path. An implicit calculation of the solution and its derivatives is then made at each stage of the calculation. This procedure can overcome some of the issues associated with the CFL constraint. The ability to stably take long time steps, along with a high order of accuracy for the advective terms, has led to the SISL method being widely used for NWP, including the dynamical cores of models at the Met Office [WSW$^+$14], the European Centre for Medium-Range Weather Forecasts [THA01], the China Meteorological Administration [CXY$^+$08] and the Canadian Meteorological Centre [CGM$^+$98], among others. However, the use of SISL methods requires the calculation, at each time step, of the departure (or arrival) points of the semi-Lagrangian paths, and the interpolation of the solution described on the computational mesh, onto these new points. The calculation of these departure points, and the subsequent interpolation of the solution, leads to additional errors in the calculation of the solution. These errors, particularly those due to the interpolation error, are greatly exacerbated when the solution changes rapidly (as in the case of a front) over a small region in space and can be compounded when a small time step Δt is used, requiring more interpolation steps. This motivates the use of an adaptive mesh method for such problems.

In this paper we will derive a novel location based r-adaptive moving mesh method **MMSISL** that couples a location based method for finding the mesh points with the SISL algorithm described above. In this algorithm we will have (a fixed number of) N_x mesh points that move as the solution evolves; in particular the mesh points will be concentrated in regions at which the computed solution has a high curvature. Location based moving mesh methods work by prescribing the mesh point location through its spatial density, and differ from Lagrangian methods which move mesh points at a prescribed velocity. They have the advantages of avoiding mesh tangling problems and of being able to resolve local solution features without denuding points from other areas of the domain [HR10]. Such methods have been developed for problems both in one-

dimension, as in this paper, and also in higher dimension and have proved their potential for many such applications, showing themselves highly competitive with other adaptive procedures. For examples of a variety of different moving mesh methods see [HR10, BHR09, TT03]. However a significant problem with using spatially adaptive methods has been the stability restriction on the time step imposed by applying the CFL condition to a (locally adapted) small spatial mesh, see [HR10, p. 99]. In particular as the adaptive mesh size becomes small, so must the time step, and this can be a significant restriction on the use of the method. Coupling a location based method to a SISL time stepping approach however presents the attractive possibility of obtaining the good local resolution of the adaptive method but without the CFL imposed time step restriction. In this paper we will make a first step in realising this objective for a one-dimensional system by developing the **MMSISL** algorithm. We will do this using a location based method to find the *arrival points* of the SISL algorithm. This will be done by prescribing the *density* of the arrival points by equidistributing an appropriate monitor function, typically based upon the solution curvature. Such moving mesh methods are known to be effective in reducing interpolation errors [HR10] and also in avoiding the mesh tangling. They are thus very appropriate to be used in the context of a SISL method where, as we will demonstrate, interpolation is an important part of the accuracy of the calculation. The coupling of moving mesh methods to calculate the arrival points with SISL methods is very natural, as the interpolation step usually encountered in applying moving mesh methods, which is a common source of difficulty and cost in the use of such methods [TT03], is *treated automatically* in the SISL framework at *little extra computational cost*. We demonstrate for Burgers' equation, and other one-dimensional PDE problems, such as the Shallow Water Equations, and the Sod gas tube problem, that the **MMSISL** algorithm gives a markedly better resolution of sharp moving features than a fixed mesh. Furthermore, the **MMSISL** method has none of the CFL related stability problems associated with other moving mesh methods, allowing significantly larger time steps than these methods.

1.2. Summary of the paper

The layout of the remainder of this paper is as follows: In Section 2 we review and describe the basic SISL method for one dimensional advective equations and briefly discuss some existing error analysis of these methods. In Section 3 we then combine the SISL strategy with a moving mesh method to derive the MMSISL algorithm. In Section 4 we apply the method to a series of challenging problems including Burgers' equation with small viscosity and the Sod gas tube problem, and we compare this method to a fixed mesh method and an alternative (non SISL based) moving mesh method. Finally in Section 5 we draw some conclusions from this work and consider future extensions of the methods used.

2. The SISL discretisation

The SISL method is the combination of a semi-implicit (SI) time stepping scheme, to remove any stability constraints from fast waves, along with a semi-Lagrangian (SL) method to provide an accurate and stable representation of the advective terms. For a semi-implicit scheme, following [SC91], only the terms that correspond to fast but physically unimportant waves (such as acoustic waves in the atmosphere) are treated implicitly, so that the overall scheme is not restricted by any CFL condition on waves that carry little physical relevance. Within both parts of the SISL method there are a number of flavours, such as using two- or three-time-level schemes and the method chosen to compute the semi-Lagrangian trajectories. Here we will concentrate on the form of SISL used in the Met Office model [WSW+14].

2.1. Overview

In this section, we will describe the popular *two time-level SISL* algorithm used to approximate the solution of a nonlinear advection dominated one-dimensional partial differential equation (PDE), which we assume has the form

$$\frac{Du}{Dt} = F\left(u, u_x, \varepsilon u_{xx}\right), \quad x \in [a, a+L], \quad t > 0, \quad \varepsilon \ll 1, \tag{1}$$

where

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + u\frac{\partial}{\partial x}$$

is the advective derivative. The basic philosophy behind all semi-Lagrangian methods is to find approximations for the particle paths of the solutions of (1), and to then calculate an approximation to the advective derivative along these paths. Integrating (1) along a semi-Lagrangian trajectory, from the departure point X_D at t to the arrival point X_A at $t + \Delta t$ over a single time step, we obtain

$$u^{t+\Delta t}\left(X_A\right) - u^t\left(X_D\right) = \int_t^{t+\Delta t} F\left(u, u_x, \varepsilon u_{xx}\right) \mathrm{d}t', \tag{2}$$

where subscripts A and D denote evaluation at arrival and departure points respectively. We will assume that at the time level $t^n \equiv t^0 = N\Delta t$ there is a spatial mesh X_j^n and that we have the approximation $u\left(X_j^n, t^n\right) \approx U_j^n$. There are then two main forms of semi-Lagrangian methods.

For a *forward semi-Lagrangian method* we depart from the given mesh X_j^n at time t^n and calculate an arrival mesh at time level t^{n+1} from the advection; the fields are then interpolated back onto the given mesh X_j^{n+1} for the next SL step. In this case, departure points are the known mesh points $X_{j,D} \equiv X_j^n$, and the solution at the arrival points has to be computed via interpolation.

In this paper we will adopt a different approach, and shall concentrate on the *backwards semi-Lagrangian method* which is widely used in the meteorological

literature, for example in the ENDGame code [WSW+14]. For this method, the arrival points $X_{j,A}$ are exactly the mesh points X_j^{n+1} at the time level t^{n+1}, and the departure points at the time level t^n are those points for which the Lagrangian paths starting at $X_{j,D}$ at time t^n arrive at $X_{j,A}$ at time t^{n+1}. Accordingly, let us assume that at time level t^n we have an approximate solution U_j^n given at all of the mesh points X_j^n We then seek an approximation to the solution at time level t^{n+1}. In a Lagrangian method we find approximations to the particle paths which satisfy the equation

$$\frac{Dx}{Dt} = u \tag{3}$$

and discretise the solution along these. To complete the SISL discretisation, the right-hand side of (2) is approximated by

$$U^{n+1}\left(X_{j,A}\right) - U^n\left(X_{j,D}\right) = \Delta t\left[\theta F^{n+1}\left(X_{j,A}\right) + \left[1-\theta\right] F^n\left(X_{j,D}\right)\right] , \tag{4}$$

(where $0 < \theta \leq 1$) and the dependencies of F have been dropped for convenience. The semi-implicit method will treat the terms that contribute to the fast waves in F^{n+1} implicitly, and the remaining terms are approximated in some way, such as extrapolation. If the fast wave contribution from F is $\mathcal{L}$ (and the remaining terms are $F-\mathcal{L}$) then (4) with a SISL discretisation becomes

$$\begin{aligned}\left[U^{n+1} - \theta\Delta t\mathcal{L}^{n+1}\right]\left(X_{j,A}\right) = \left[U^n + \left(1-\theta\right)\Delta t F^n\right]\left(X_{j,D}\right) \\ + \theta\Delta t\left(F-\mathcal{L}\right)^{n+1}\left(X_{j,A}\right) .\end{aligned}$$

Treating (3) in the same way yields

$$X_{j,A} - X_{j,D} = \Delta t\left[\theta_X\ U^{n+1}\left(X_{j,A}\right) + \left(1-\theta_X\right)\ U^n\left(X_{j,D}\right)\right] , \tag{5}$$

where $0 < \theta_X < 1$. For this paper we shall take $\theta = 1/2$ for all of the calculations. The discretisation is completed firstly by choosing the method of interpolation from the solution at the mesh points onto the departure points of the Lagrangian paths, and secondly choosing an appropriate spatial discretisation of the (derivatives of the) approximation on the (possibly non-uniform) mesh X_j^n.

Following [WSW+14] the departure points $X_{j,D}$, according to (5), are calculated iteratively, with the $u^n\left(X_D\right)$ term lagged in the iterates. Observing the change in the departure points over these iterates, we see this process rapidly converges; we find 10 iterates to be sufficient to reach convergence for all time steps. The solution at time level t^n is then interpolated onto these departure points. It is known [McC88, McD84, McD87] that both the calculation of these points and the interpolation of the solution onto these points can lead to (possibly large) errors. These errors can be significantly reduced (particularly in the context of a moving mesh), if the arrival and departure points are both close to the computational mesh. We will address this in Section 3. The choice of the

underlying spatial discretisation, made in this paper, follows the ENDGame formulation using centred finite differences. However, in principle there is no barrier to using other forms of discretisation, such as a spectral [THA01] or Galerkin [GNQ14] method.

Interpolation to the departure points For a positive CFL number the departure point $X_{j,D}$ lies between two mesh points so that

$$X^n_{j-\mathcal{N}-1} < X_{j,D} \leq X^n_{j-\mathcal{N}}$$

where $\mathcal{N}$ is the integer part of the CFL number. An important part of the semi-Lagrangian method is the interpolation step, where we project the solution onto the departure points. In operational codes, various methods of interpolation are used, most commonly some form of Lagrangian polynomials. The simplest of these is *linear interpolation*, so that

$$U_{j,D} = yU^n_{j-\mathcal{N}-1} + (1-y)U^n_{j-\mathcal{N}}\,, \quad y = \frac{X^n_{j-\mathcal{N}} - X_{j,D}}{X^n_{j-\mathcal{N}} - X^n_{j-\mathcal{N}-1}}\,.$$

This scheme has the advantage of being monotone, however it only first order accurate. In practice, higher order methods are often used instead. Of these, cubic-Lagrange interpolation is a popular choice, providing a good balance between accuracy and computational cost. The cubic-Lagrange interpolant of a function $f(x)$ for an interval $[x_i, x_{i+1}]$ is the unique cubic which passes through points f_{i-1}, f_i, f_{i+1} and f_{i+2}, where $f_j = f(x_j)$. Alternatively, a number of other choices could be made, such as to use the essentially non-oscillatory (ENO) cubic interpolant [HEOC87], or weighted essentially non-oscillatory (WENO) interpolants [LOC94]; a cubic Hermite interpolant can be used which is uniquely determined on $[x_i, x_{i+1}]$ by the points f_i, f_{i+1} and the gradients d_i, d_{i+1}, where $d_j = f'(x_j)$ (determined numerically). A cubic Hermite interpolant can be guaranteed to be monotonic by ensuring the gradients sit within a specific range [FC80, dBS77]. Rasch and Williamson [RW90] compared a number of different interpolants and derivative estimates for semi-Lagrangian advection, and found some of the best all-round results with a monotonic cubic Hermite interpolant using a fourth-order finite difference estimate for the derivatives [Hym83].

2.2. The nature of the errors encountered when using SISL methods

The accuracy of the SISL algorithm has been studied by a number of authors in the context of fixed (and uniform) meshes: see for example [MM94, CK09, Smi00, McC88, McD84, McD87]. Of most relevance to the current paper is the observation in [McC88, McD84, McD87], that the interpolation step in the SISL introduces an error which takes the form of a dissipative change to the underlying solution. In these papers the dissipation factor for plane waves is calculated for a set of different interpolation strategies and fractions of the CFL number. In all cases a significant increase in the dissipation was observed, often inversely proportional to the time step Δt. We also demonstrate through the Burgers' equation example studied in this paper that there is a further dispersive

error which manifests itself through an incorrect estimation of the speed of the moving sharp front.

3. The Moving Mesh SISL (MMSISL) Method

3.1. Overview

In this section we develop the moving mesh SISL algorithm **MMSISL**. As remarked above, one of the main sources of error in applying the SISL method is in the interpolation step when the solution is projected from the grid points onto the departure points. Such an error will be larger when the solution is evolving rapidly, for example at a front or interface. Hence some form of locally adaptive spatial step size control close to the front is attractive. To implement this we will consider an r-adaptive moving mesh method for finding the points X_j^n. In particular we will use a location dependent approach in which the positions of the mesh points are determined by prescribing the local *density* of the mesh. This approach is different, both in philosophy and practice, from the Lagrangian approach of prescribing the velocity of the mesh points, typically to be the same as that of the solution. This latter approach is known to lead to skew and tangled meshes. In contrast the location based mesh methods (for problems posed in several dimensions) are known to lead to regular meshes which avoid tangling [HR10, Zeg07, TT03, BHR09]. Location based methods have proven themselves in many applications [TT03, BHR09], in particular when applied to an anelastic atmospheric flow solver in [KSD12] (which used a flux-form discretisation), in calculating the developing fronts in tropical storm problems [BCW13], in shock dynamics for hyperbolic problems [FL03], and in data assimilation for meteorological problems [PC11]. They also have the advantage over adaptive mesh refinement methods (AMR), of keeping the same number of mesh points at each time step, simplifying the computational implementation. However, a significant problem with the use of such methods is that the small spatial mesh required to resolve the local features, often means that an equally small time step must be taken due to the CFL condition. It is thus attractive to consider coupling them to a SISL algorithm where no such restriction applies.

This approach has an additional benefit. As we have seen in the previous sections, the principle cause of both the diffusive and the dispersive errors in the use of the SISL algorithm is the process of interpolating the function defined on the mesh onto the departure points. However, in [HR10] it is shown that a location based adaptive mesh method can minimise the interpolation error when approximating a function if the density of the mesh points is *proportional to the local curvature of the function.* Thus we can expect to reduce the errors of the SISL algorithm substantially when using such an approach, with the coupling to SISL meaning that we avoid the CFL restriction. Even better, we show that this can be done by making relatively small changes to the SISL algorithm itself.

Accordingly, in this section we develop a one-dimensional moving mesh strategy **MMSISL** based on equidistribution of the estimated spatial interpolation error

at each time step. This will then be coupled to the SISL method outlined in Section 2 by using the location based moving mesh method to update the location of the arrival points at each time step. We will demonstrate by a series of numerical tests that this gives a much better resolution of solutions with moving fronts than fixed mesh methods, even when higher order interpolation methods are used.

3.2. Moving mesh calculations

3.2.1. Overview of the location based equidistribution method

Suppose that we wish to approximate a continuous function $u(x)$ with $x \in [a, b]$ on a non-uniform spatial mesh $X_j \in [a, b]$ with $j = 0 \dots N$, where N is fixed; we are free to place the mesh points by moving X_j during the course of the calculation. Suppose further that $\rho(x)$ is a monitor function which is an estimate of the error of the resulting approximation and which controls the mesh density. An effective location based procedure for placing the points to minimise the error over all possible meshes, is to *equidistribute* the monitor function so that there is a constant Θ given by

$$\int_a^b \rho(x)\, \mathrm{d}x = \Theta\,,$$

and the mesh satisfies the condition

$$\int_{X_j}^{X_{j+1}} \rho(x)\, \mathrm{d}x = \frac{\Theta}{N}\,. \tag{6}$$

This method was introduced in [dBS77] and is described in detail in [HR10]. If we wish to minimise the *interpolation error* when approximating $u(x)$ with piecewise linear functions, it is shown in [HR10] that we should take the error measure $\rho(x)$ to be a normalised scaling of the *curvature* of the underlying function, so that

$$\rho(x) = \sqrt{1 + \beta^2 u_{xx}^2}\,, \tag{7}$$

where the constant β is chosen both to balance the number of mesh points placed in the region where the solution changes rapidly, with the number of points placed elsewhere, so that both regions are well resolved, and also to enforce the degree of adaptivity required. Trivially $\beta = 0$ corresponds to a uniform mesh. The mesh points X_j can then be calculated from the expression (6) by quadrature. We compare the use of this 'optimal for linear interpolation' monitor function with the commonly used *arc-length monitor function* given by

$$\rho(x) = \sqrt{1 + \beta^2 u_x^2}\,. \tag{8}$$

In practice, as we will see, the arc-length monitor function can give more stable meshes.

In the SISL calculation, the function $u(x, t)$ is not known a-priori but is approximated as part of the solution. However, we may implement a two-stage

strategy to calculate a moving mesh X_j^n at each time level t^n which makes use of the expression (6). We note that many moving mesh strategies (see for example [TT03]) need to interpolate the solution onto the new mesh once it has been calculated. This step is not required in the algorithm we propose, as the interpolation of the solution onto the departure points in the SISL algorithm automatically deals with this issue.

3.2.2. The coupling of the location based moving mesh and SISL algorithms. The **MMSISL** algorithm for coupling a moving mesh method to the SISL algorithm is developed in a two-step process as follows:

1. At time level t^n we have a computed solution U_j^n and a computed arrival mesh $X_{j,A}^n$. At the time level t^0 the solution U_j^0 is the value of the initial state, and the mesh X_j^0 is assumed to be *uniform*.
2. Using the known values of U_j^n, a new arrival mesh $X_{j,A}^{n+1}$ is calculated to equidistribute the monitor function $\rho(x)$ evaluated on U_j^n.
3. Using the new arrival mesh $X_{j,A}^{n+1}$, a new solution U_j^{n+1} is calculated at these points using the SISL algorithm as described in Section 2 with appropriate calculation of the departure points X_D, and interpolation onto these points. In implementing this we observe that all of the spatial derivatives must be calculated carefully on the *non-uniform mesh* X_j^{n+1} to the desired order of accuracy.

We note that this procedure is essentially that of constructing a mesh at time t^{n+1} which equidistributes the monitor function of the solution at time t^n.

The main new part of this computation which differs from the earlier SISL calculation is the evaluation of the new arrival points $X_{j,A}^{n+1}$ in Step 2. This is done as follows:

1. Using the values of U_j^n on the current arrival points $X_{j,A}^n$, the monitor function ρ_j^n is evaluated at each point $X_{j,A}^n$ from either the arc-length or the curvature functions by using a finite difference approximation.
2. The resulting point values ρ_j^n of the monitor function are then *smoothed* using a low pass filter. An example of this is to use several passes of a five point moving average filter. This gives new values $\widehat{\rho}_j^n$. The smoothed function $\widehat{\rho}$ is then used to equidistribute the mesh points. This smoothing has proved necessary in earlier calculations of moving meshes (see for example [PC11, HR10]) to ensure that we have a regular mesh at each stage of the calculation.
3. A linear spline is used to reconstruct a continuous monitor function $\rho(x, t_n)$ from the smoothed points $\widehat{\rho}_j^n$.
4. Provisional values of the new arrival points $X_{j,A}^*$ are then calculated from the equidistribution equation (6). In this calculation, the linear spline

function is integrated exactly to give a piecewise quadratic function. The points $X^*_{j,A}$ are then found by solving a locally quadratic equation.

5. The new arrival points are obtained through the relaxation condition

$$X^{n+1}_{j,A} = \lambda X^*_{j,A} + (1-\lambda) X^n_{j,A}\,, \tag{9}$$

where $0 < \lambda \leq 1$ is chosen appropriately.

NOTE The final step in this algorithm is necessary to give a stable and regular mesh if the mesh points would otherwise evolve too rapidly. Other methods which employ relaxation and which are based on moving mesh PDEs, are described in [HR10].

The advantage of using a moving mesh instead of a fixed mesh, is that the underlying function is better resolved by the adapted mesh, with lower interpolation errors. Indeed, in a well designed moving mesh method we expect that locally the mesh spacing close to a shock should of the order of the shock width, and if a piece-wise linear interpolant is used then the interpolation errors will be minimised by using the curvature monitor function (7) [HR10]. We will now demonstrate through some numerical experiments that these considerations lead to an accurate, and robust numerical approximation method.

3.3. Extensions and limitations of the MMSISL method

The MMSISL method as described is only appropriate for one-dimensional problems. However, this is only meant to be a first study for this method. It is straightforward to extend location based r-adaptive method to two and three dimensional problems, see for example the various methods described in [HR10], including variational, optimal transport, and moving mesh PDE based methods. Such methods are known to deliver effective meshes without mesh tangling issues. There are no reasons in principle why these established methods could not be combined with a higher dimensional SISL method using similar ideas to those developed in this section. However there are naturally many technical issues that need to be resolved, and these methods need then careful comparison with other techniques for mesh adaption. We leave this as a subject for future work.

4. Numerical results and comparisons

4.1. Overview

We now test the MMSISL method on a number of challenging examples in one-dimension and make comparisons with both fixed mesh methods and non SISL based adaptive methods. We choose examples which develop sharp solution features over small length scales.

4.2. Application to Burgers' equation with small viscosity and a single travelling shock

We consider the viscous Burgers' partial differential equation

$$u_t + uu_x = \varepsilon u_{xx}, \quad 0 < \varepsilon \ll 1\,, \tag{10}$$

with asymptotic boundary conditions given by

$$u(-\infty, t) = c + \alpha,\ u(\infty, t) = c - \alpha\,.$$

This problem has the exact, asymptotically stable, travelling wave solution with a single shock given by

$$u^*(x,t) = c - \alpha \tanh\left(\frac{\alpha(x-ct)}{2\varepsilon}\right)\,.$$

This travelling wave is a prototype of frontal behaviour encountered in meteorology and of shock formation encountered in gas dynamics [LeV02a]. A key feature of this solution is the existence of a sharp front of width $\delta = \varepsilon/\alpha$ and front gradient of

$$u_x(x^*, t) = -\alpha^2/2\varepsilon\,.$$

To fully resolve this profile requires a spatial step size Δx such that

$$\Delta x < \delta\,.$$

We now compare the behaviour of the fixed mesh SISL scheme, using two forms of interpolation, with that of the MMSISL method.

4.2.1. Fixed mesh SISL approximation

Applying the SISL discretisation with $\theta = 1/2$ to Burgers' equation (10) and using a uniform arrival mesh of fixed spacing Δx gives the numerical scheme

$$\left(U_j - \frac{\Delta t}{2}\varepsilon\ \delta_{xx}U_j\right)_A^{n+1} = \left(U_j + \frac{\Delta t}{2}\varepsilon\ \delta_{xx}U_j\right)_D^{n} \tag{11}$$

along with the trajectory equation (5) with $\theta_X = 1/2$ and centred finite difference approximations of the spatial derivatives.

We firstly take a *piecewise linear interpolant* and perform an experiment in which

$$\varepsilon = 10^{-4}, \quad \alpha = 0.1, \quad \Delta x = 0.05, \quad \Delta t = 0.0375.$$

Observe that $\delta = 10^{-3} \ll \Delta x$. We approximate the infinite spatial domain by the interval $[-5, 5]$ and take $\alpha = 0.1$ and $u(-5, t) = 1 + \alpha, u(5, t) = 1 - \alpha$, so that the wave speed is $c = 1$ with CFL number $\nu_{\mathrm{CFL}} = 0.75$. The resulting x-profiles of the numerical approximation U to $u(x, t)$ are given in Figure 1 for a series of values of increasing time t. We next take $t = 1.5$ and in Figure 2

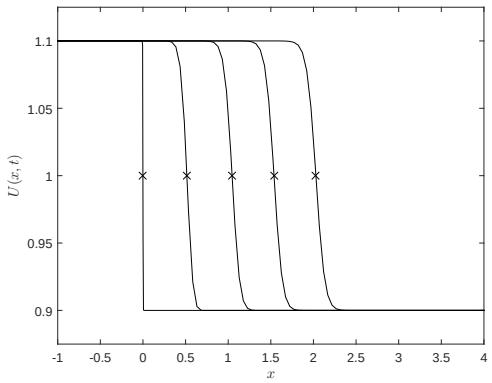


Figure 1: Initial condition and the numerical solution U to the SISL discretisation of Burgers' equation. Shown moving to the right are the initial condition at $t = 0$, and solutions at times $t = 0.5, 1, 1.5$ and 2. The crosses correspond to the centres of the front x^* such that $U(x^*) = c = 1$.

(left) we compare the numerical solution U with the exact solution u with $c = 1$ in a vicinity of the front. The figure shows that the numerical solution has a broader front than the exact solution (a diffusive contribution to the error). Furthermore the centre of the front of the numerical solution at $x \approx 1.575$ is well in advance of that of the exact solution which is at $x = 1.5$ (a dispersive contribution to the error). In Figure 2 (Right) we repeat this calculation using

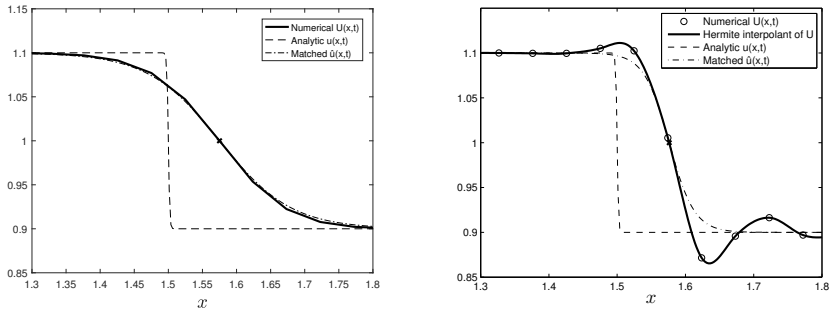


Figure 2: Comparison of the numerical solution with linear interpolation (solid line), with the true solution (dashed line) at the time $t = 1.5$. (Left) *Linear interpolation* It can be seen that the front for the numerical solution is ahead of that of the analytical solution and is significantly broader. (Right) *Non monotone cubic-Hermite interpolation.* The solution is more oscillatory than before. However, as before, the front for the numerical solution is ahead of that of the analytical solution and is also significantly broader.

a *non monotonic cubic-Hermite interpolant without a flux limiter* and compare the true and numerical solution at the time $t = 1.5$. As might be expected this method leads to a solution with more oscillations than those with linear

interpolation due to the non-monotone nature of the interpolation scheme used. However, many similar features to the earlier calculations remain. In particular the front has also broadened and has travelled at a higher speed than the true solution.

We next make a direct comparison of the numerical and the true solution by computing the L_2 solution error $E = \|u^* - U\|_2$ as a function of $\Delta x \equiv 10/N_x$ at $t_{\max} = 2$, for three values of $\Delta t = 0.01, 0.02, 0.04$. The results for the two interpolation schemes are presented in Figure 3. Observe both the slow reduction of the maximum value of the error and also its oscillatory nature (which arises at values of N_x when the semi-Lagrangian paths cross mesh points and the CFL number takes integer values). An analysis of these errors is made in [**?**]

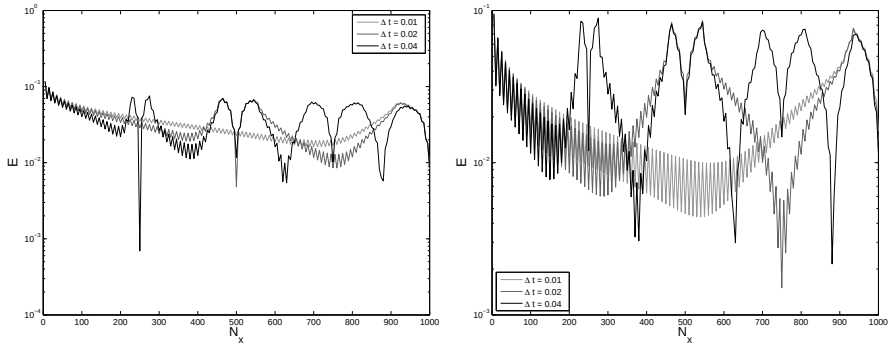


Figure 3: The L_2 solution error E of the fixed mesh SISL scheme, at $t_{\max} = 2$, as a function of $\Delta x = 10/N_x$, for $\Delta t = 0.01, 0.02, 0.04$ (Left) Linear interpolation (Right) non monotonic cubic-Hermite interpolant without a flux limiter.

4.2.2. The MMSISL approximation

For comparison we apply the MMSISL algorithm to the same problem. The discretisation in (11) is used with the spatial derivatives now approximated by the difference operator

$$\delta_{xx} U_j \equiv \frac{(U_{j+1} - U_j)/(X_{j+1} - X_j) - (U_j - U_{j-1})/(X_j - X_{j-1})}{(X_{j+1} - X_{j-1})/2} .$$

As a first calculation, we take the *arc-length* monitor function (8) with $\beta = 30$ and applying four iterations of a five point moving average smoothing filter to the resulting monitor function. We also take the relaxation factor $\lambda = 0$ in (9). We take the interval $x \in [-5, 5]$ and $t \in [0, 2]$ and take $N_x = 60$ so that in the uniform mesh case $\Delta x = 10/N_x$. We take $\Delta t = 0.04$. We plot both the evolution of the solution over the interval $x \in [-5, 5]$ compared to the true solution at $t = 2$. We also plot the trajectories of arrival points. We can see

that the sharp solution front of the true solution is preserved and is followed closely by the moving mesh points.

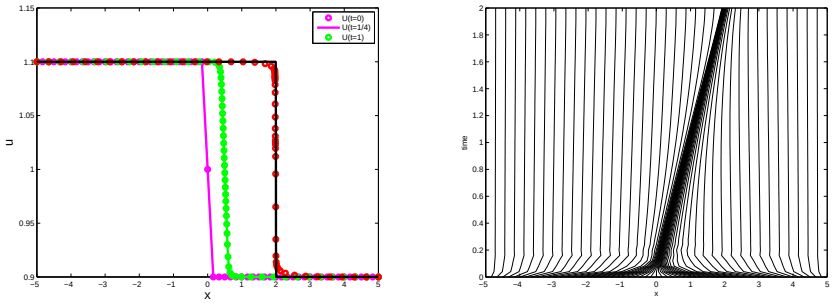


Figure 4: (a) The evolving solution showing the sharp front and the close agreement between the solution on the adapted mesh at time $t = 2$ and the true solution. (b) The moving mesh point trajectories which closely follow the front.

As a second calculation, in Figure 5 we plot the L_2 solution error E as a function of N_x for $\Delta t = 0.04$ and varying values of $\beta = 0, 10, 30, 50$. In the adaptive case when $\beta > 0$ there is a maximum value of N_x at which point the adaptive mesh method becomes unstable. We see a very significant improvement in the L_2 solution error over the calculations using a fixed mesh (with $\beta = 0$). Over this range the error in the uniform mesh calculation decreases very slowly. In contrast the error in the moving mesh calculation decreases very rapidly with N_x up to the point of instability. The rate of decrease in the error is larger for larger values of β and the maximum value for N_x is smaller. In the optimum choice arises when $\beta = 30$ and $N_x = 60$. At this value the minimum mesh spacing is $\Delta x_{min} = 0.0035$ and the maximum $\Delta x_{max} = 0.297$. Hence the maximum CFL number is $\nu_{CFL} = 0.04/0.0035 = 11.5$. However the use of the SISL method has avoided the instabilities associated with such a high CFL number.

Similar results arise when a curvature monitor function is used, however this is more unstable than the arc-length monitor.

We conclude that the use of a location based moving mesh methods coupled to SISL significantly improves the resolution of the front of Burgers' equation without either adding much extra cost to the calculation, or having a CFL restriction on the time step.

4.2.3. Comparison with an MMPDE adaptive mesh method, of the solution of Burgers' equation with two shocks

In the text book on adaptive moving mesh methods [HR10, §1.4], a similar study to that above is made of a solution of Burgers' equation (with $\epsilon = 10^{-4}$) with

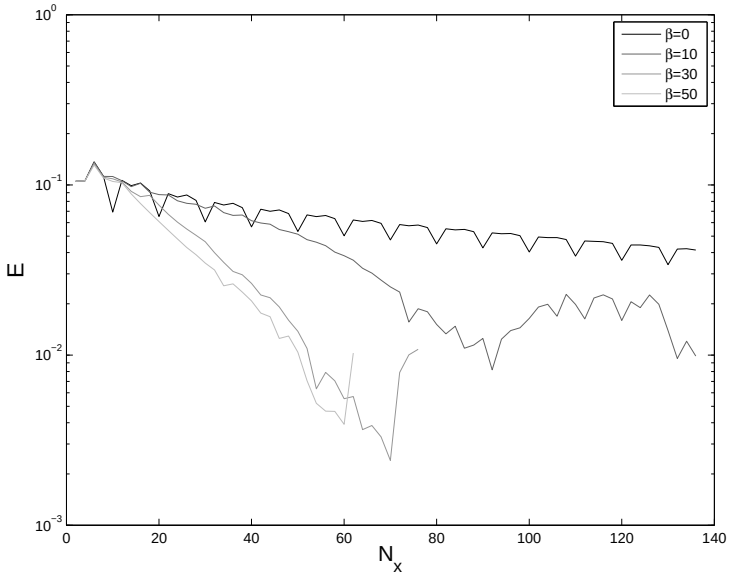


Figure 5: The L_2 solution error E plotted as a function of N_x for $\Delta t = 0.04$ and varying values of β including the uniform mesh case of $\beta = 0$ (black). In this case we use a moving mesh with an arc-length monitor function.

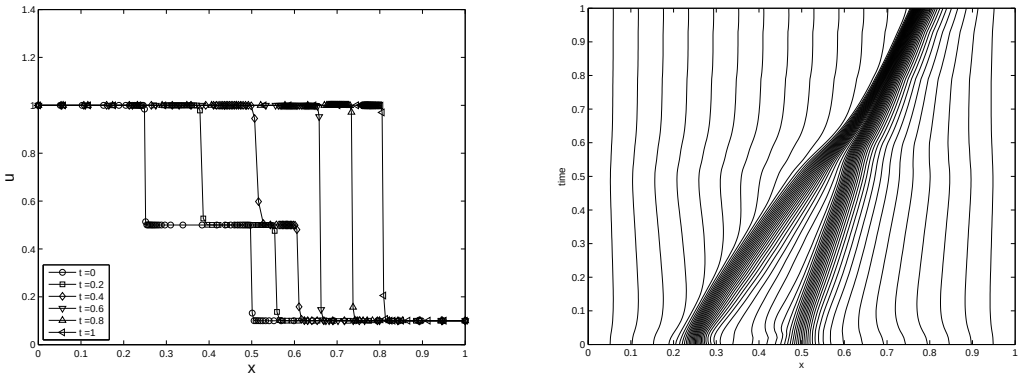
two evolving sharp fronts which coalesce. This takes the exact form

$$u(x,t) = \frac{0.1e^{\frac{-x+0.5-4.95t}{20\epsilon}} + 0.5e^{\frac{-x+0.5-0.75t}{4\epsilon}} + e^{\frac{-x+0.375}{2\epsilon}}}{e^{\frac{-x+0.5-4.95t}{20\epsilon}} + e^{\frac{-x+0.5-0.75t}{4\epsilon}} + e^{\frac{-x+0.375}{2\epsilon}}}.$$

In this problem the solution has two shocks which coalesce and then move as a single shock. We now use this as the second example for the MMSISL method. The numerical calculation in [HR10] was made by using a moving mesh PDE (MMPDE) method, which combines equidistribution with a PDE based relaxation algorithm to move the mesh points. In this approach Burgers' equation is put into Lagrangian form (to allow for the mesh movement) and then discretised in space using a finite difference method with $N_x = 61$ mesh points on a non-uniform mesh. The MMPDE method also used a curvature monitor function. The resulting ODEs are then solved using the backward differentiation formula of order 5 in the Matlab ODE solver `ode15i`. In the calculation presented in [HR10] the H^1 solution error scales as $1/N_x$. However, the time steps taken are much smaller than in the SISL method. The resulting values of the time step used by the solver are presented in Figure 1.8 of [HR10] and are revealing. In many cases a time step $\Delta t < 10^{-4}$ had to be used by the solver to give a stable solution for the general case of the travelling shocks. At the time when the two shocks coalesce the time step reduces rapidly and the value of $\Delta t = 2 \times 10^{-6}$ was used over a short interval. This is generally consistent with having a time step limited by the CFL condition applied to the smallest mesh size which is of

order 10^{-4}.

We contrast these results with the application of the MMSISL method using an arc-length monitor. In this case we take $N_x = 60$ and the fixed time step of $\Delta t = 0.01$. Because the fronts are much larger in amplitude than the previous example we take the smaller value of $\beta = 2.3$. Similarly, to allow the mesh to cope with the coalescence of the two shocks and remain stable we use a relaxation parameter of $\lambda = 0.1$ in (9). The resulting calculation is shown in Figure 6. We can see that the MMSISL scheme has accurately followed the two fronts and the coalescence of the front. We observe that this has been achieved with with a fixed time step of $\Delta t = 10^{-2}$, which is significantly larger than that used in [HR10]. For comparison, the same calculation but with a fixed mesh ($\beta = 0$) is given in Figure 7.



(a) (b)

Figure 6: Evolution of the (a) u field at different values of $t = 0.0.2, 0.4., 0.6, 0.8, 1$ and (b) the moving mesh, in the double shock case of [HR10] using a time step of $\Delta t = 10^{-2}$ and the MMSISL method.

4.3. Nonlinear Shallow Water Model

We now consider a number of test cases in which the MMSISL method is applied to systems of PDEs. The non-rotating nonlinear shallow water equations in one spatial dimension take the form

$$\frac{Du}{Dt} + g\frac{\partial h}{\partial x} = 0, \tag{12}$$

$$\frac{Dh}{Dt} + h\frac{\partial u}{\partial x} = 0, \tag{13}$$

for the height field $h(x,t)$ and the velocity field $u(x,t)$, where g is the acceleration due to gravity. Applying the semi-implicit semi-Lagrangian discretisation on a

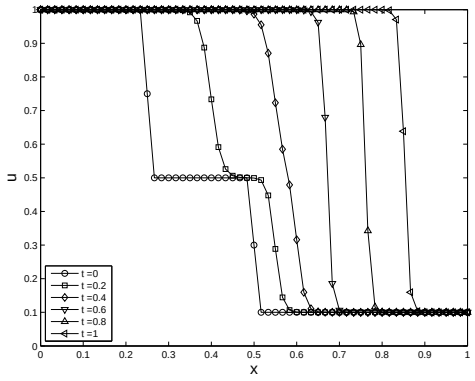


Figure 7: Evolution of the u field with a uniform mesh using a time step of $\Delta t = 10^{-2}$.

staggered grid to (12)-(13) gives

$$\left(u + \theta \Delta t g \frac{\partial h}{\partial x}\right)_A^{n+1} = \left(u - (1-\theta)\,\Delta t g \frac{\partial h}{\partial x}\right)_D^{n}, \tag{14}$$

$$\left(h + \theta \Delta t h \frac{\partial u}{\partial x}\right)_A^{n+1} = \left(h - (1-\theta)\,\Delta t h \frac{\partial u}{\partial x}\right)_D^{n}. \tag{15}$$

along with the kinematic equation (5). This is solved using an iterative semi-implicit method for the increments $(\delta h,\, \delta u) \equiv \left(h^{(k+1)} - h^{(k)},\, u^{(k+1)} - u^{(k)}\right)$ where $\left(h^{(k)},\, u^{(k)}\right)$ are the k^{th} estimate of the iterative scheme for the time level $n+1$ fields. The system of equations solved at each iteration is then

$$\delta u + \theta \Delta t g \frac{\partial \delta h}{\partial x} = -\left(u + \theta \Delta t g \frac{\partial h}{\partial x}\right)_A^{(k)} + \left(u - (1-\theta)\,\Delta t g \frac{\partial h}{\partial x}\right)_D^{n},$$

$$\delta h + \theta \Delta t h^* \frac{\partial \delta u}{\partial x} = -\left(h + \theta \Delta t h \frac{\partial u}{\partial x}\right)_A^{(k)} + \left(h - (1-\theta)\,\Delta t h \frac{\partial u}{\partial x}\right)_D^{n},$$

and h^* is a reference height field (here it is taken to be the average of the time level n height field). In all simulations presented in this section $\theta = 0.55$, there are 4 iterations of the iterative-semi-implicit method and cubic Lagrange interpolation with a monotone limiter is used for terms evaluated at the departure points. This system is used to simulate two classic problems from the literature: the Slumping Gaussian and the Dam Break problem.

4.3.1. Slumping Gaussian

In this test the initial height field is set to be a Gaussian function so that

$$h(x,0) = 1 + 0.4\exp(-5x^2)\,,$$
$$u(x,0) = u_0\,,$$

with $u_0 = 0$, $g = 1$. We take the domain to be $x \in [-5, 5]$. The initial perturbation splits into two bumps that propagate with characteristic speed

$$c = u_0 \pm \sqrt{g h_0} \equiv 1 \,.$$

As the waves propagate, due to the nonlinearity, they steepen at the front and flatten at the rear. This test is simulated using the SISL method on the above domain with $N_x = 100$ points and $\Delta t = 0.02$ and with cubic interpolation without a limiter. As the mesh movement is moderate a value of $\lambda = 1$ was used in (9). The evolution of the height field is shown in Figure 8 for both a fixed mesh (panel (a)) and a moving mesh (panel (b)) obtained by the equidistribution of a *arc-length monitor function* on the height field h with $\beta = 10$. Both the fixed mesh and the moving mesh do a good job of capturing the propagation of the waves, however the moving mesh method much better captures the steepening of the front of the waves. The evolution of the moving mesh is shown in panel (c) and the clustering of the resolution near the leading edge of the waves (and hence the improved representation of the front) can also be seen.

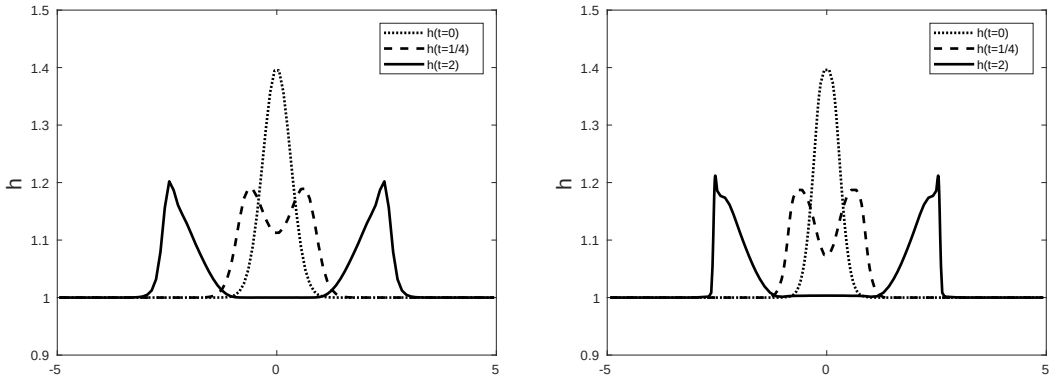
The slumping test is now further modified to include an initial background wind field $u_0 = 1$ on a domain $x \in [-2, 8]$. This results in the right most wave packet travelling twice as fast, whilst the left most packet forms a standing wave. Again, this test is run with both a static and moving mesh (with parameters as above), and the evolution of the height field is shown in Figure 9. Both methods correctly simulate the two wave packets, however the static mesh does not show much steepening of the propagating wave and the stationary wave displays a small oscillation in its tail. Both these features are much improved by using a moving mesh method, although does develop an overshoot near the leading edge of the travelling wave due to the use of the cubic interpolant. Note that there is no diffusion present in this experiment apart from the implicit diffusion in the upwind cubic interpolation used in the semi-Lagrangian scheme and the small off centring used in the semi-implicit scheme.

4.3.2. Dam Break

The dam break problem models what would happen if a barrier that initially separates a fluid at two different levels is removed, it is a special case of the Riemann problem, and is described by a piecewise constant initial height field

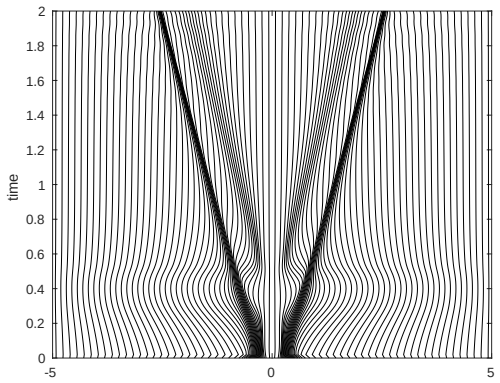
$$h(x, 0) = \begin{cases} h_l & \text{for} \quad x < 0, \\ h_r & \text{for} \quad x \geq 0, \end{cases}$$

with $h_l = 3$, $h_r = 1$ and $u(x, 0) = 0$ and the domain is $x \in [-5, 5]$. The initial discontinuity results in a rightward propagating shock wave and a leftward propagating rarefraction wave. Using the SISL method described above, the evolution of the height field is shown in Figure 10 for both a fixed mesh (panel (a)) and a moving mesh (panel (b)) obtained by equidistribution of an arc-length monitor function, with $\beta = 1, \lambda = 1$, on the height field h. In both cases $N_x = 100$ points are used and $\Delta t = 0.02$.



(a)

(b)



(c)

Figure 8: Evolution of the height field in the slumping test using (a) fixed spatial mesh and (b) moving mesh with arc-length monitor function. (c) shows the evolution of the mesh used in panel (b).

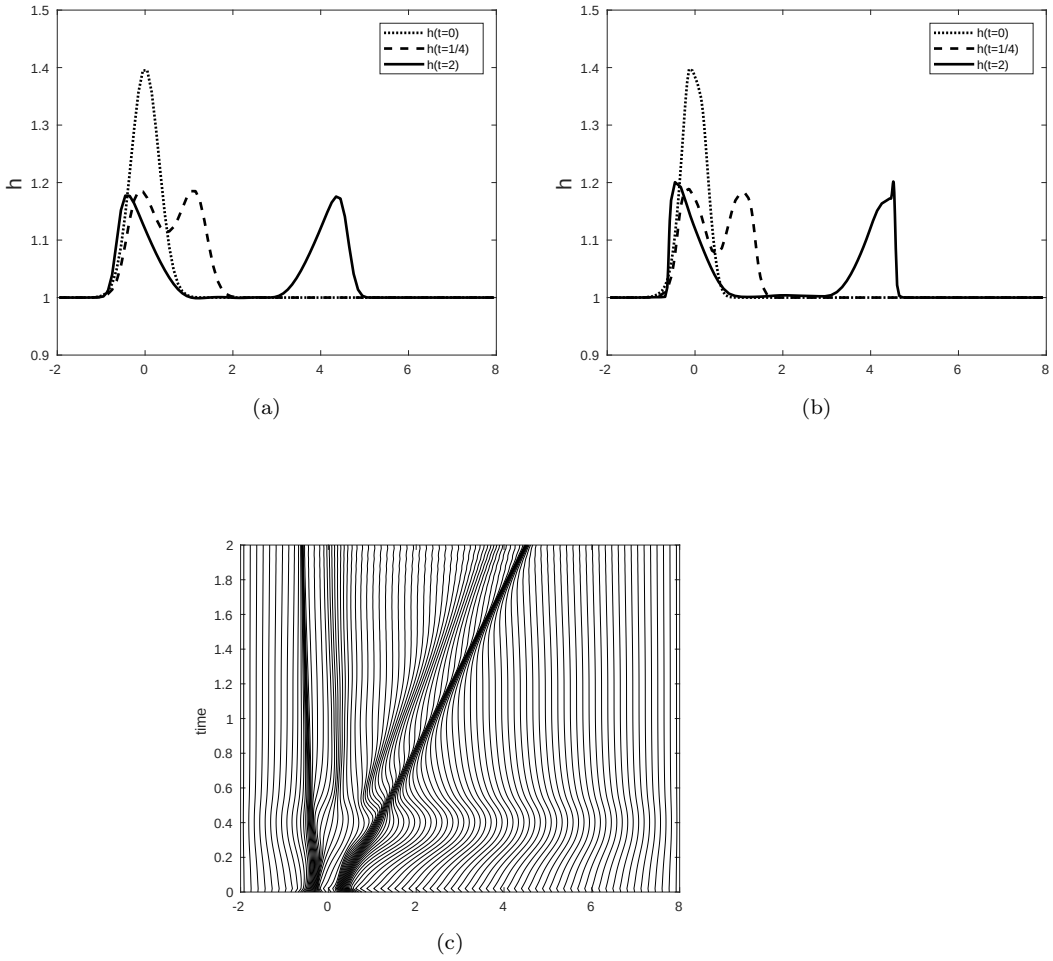
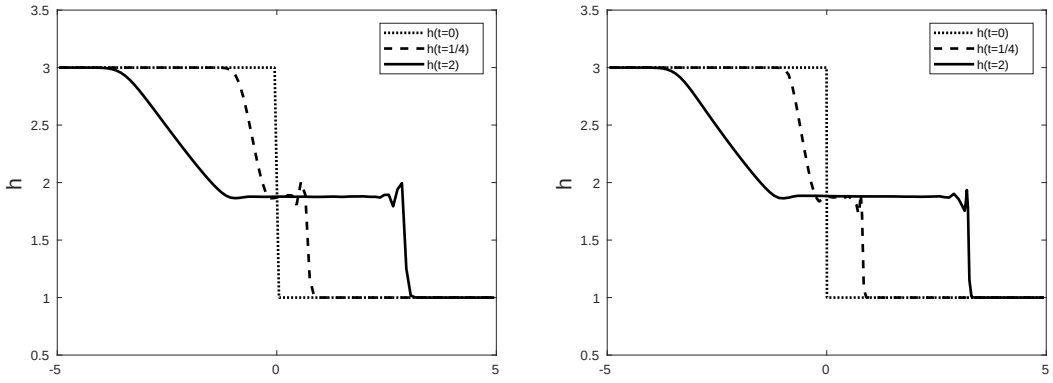
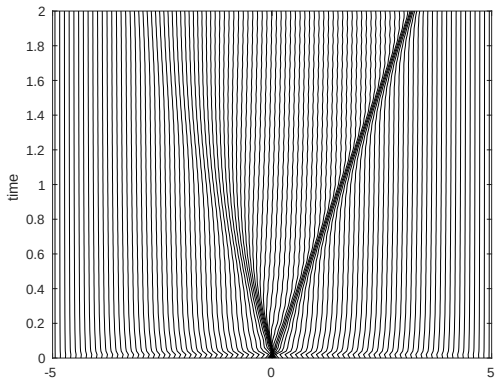


Figure 9: Evolution of the height field in the slumping test with a constant background wind field of $u = 1$ using (a) fixed spatial mesh and (b) moving mesh with arc-length monitor function. (c) shows the evolution of the mesh used in panel (b).



(a)

(b)



(c)

Figure 10: Evolution of the height field in the dam break test using (a) fixed spatial mesh and (b) moving mesh with arc-length monitor function. (c) shows the evolution of the mesh used in panel (b).

It should be noted that, in contrast with the travelling wave solutions to Burgers' equation considered earlier in this paper, these simulations consist of the fast waves that travel at characteristic speed $c = u_0 \pm \sqrt{gh_0}$ and the terms that govern this propagation are the linear terms handled by the semi-implicit scheme. This can be seen by turning off the semi-Lagrangian component of (14) and (15) (by setting the departure points equal to the arrival points) in this case the broad picture shown in Figures 8 and 10 is the same, but without any steepening of the waves.

4.4. The Sod Tube test problem.

The compressible Euler equations for gas dynamics in one-dimension can be formulated into an advective form, using variables and a formulation appropriate for atmospheric calculations [WSW+14]. In this case we consider the evolution of the velocity u, density ρ, Exner pressure Π and potential temperature Θ, with ρ, θ, Π colocated and u staggered half a grid length, through the PDE system

$$\frac{Du}{Dt} = -c_p \Theta \frac{\partial \Pi}{\partial x}, \tag{16}$$

$$\frac{D\rho}{Dt} = -\rho \frac{\partial u}{\partial x}, \tag{17}$$

$$\frac{D\Theta}{Dt} = 0, , \tag{18}$$

$$\Pi^{\frac{1-\kappa}{\kappa}} = \frac{R}{P_0} \rho \Theta. \tag{19}$$

These variable are related to the primitive variables (u, ρ, P, T) through the transformations

$$\Pi = (P/P_0)^{\kappa} \quad \text{and} \quad \Theta = \Pi / T \,,$$

where P_0 is a constant reference pressure; R is the gas constant per unit mass, c_p is the specific heat at constant pressure and $\kappa = R/c_p$.

We consider solving this system using the same two-time-level SISL method as in the previous sections. This leads to the discretisation:

$$\begin{aligned}
\left(u + \theta \Delta t c_p \Theta \frac{\partial \Pi}{\partial x} \right)_A^{n+1} &= \left(u - (1-\theta) \Delta t c_p \Theta \frac{\partial \Pi}{\partial x} \right)_D^n , \\
\left(\rho + \theta \Delta t \rho \frac{\partial u}{\partial x} \right)_A^{n+1} &= \left(\rho - (1-\theta) \Delta t \rho \frac{\partial u}{\partial x} \right)_D^n , \\
\Theta_D^{n+1} &= \Theta_D^n , \\
\left(\Pi^{\frac{1-\kappa}{\kappa}} - \frac{R}{P_0} \rho \Theta \right)_A^{n+1} &= 0 \,.
\end{aligned}$$

We solve this system of equations by using an iterative semi-implicit scheme, so

that the system solved at each iteration is given by

$$\begin{aligned}
\delta u + \theta\Delta t c_p \Theta^* \frac{\partial \delta\Pi}{\partial x} &= -\left(u + \theta\Delta t c_p \Theta \frac{\partial \Pi}{\partial x}\right)_A^{n+1} + \left(u - (1-\theta)\,\Delta t c_p \Theta \frac{\partial \Pi}{\partial x}\right)_D^{n}, \\
\delta\rho + \theta\Delta t \frac{\partial \rho^* u}{\partial x} &= -\left(\rho + \theta\Delta t \rho \frac{\partial u}{\partial x}\right)_A^{n+1} + \left(\rho - (1-\theta)\,\Delta t \rho \frac{\partial u}{\partial x}\right)_D^{n}, \\
\delta\Theta &= -\Theta_A^{n+1} + \Theta_D^{n}\,, \\
\frac{1-\kappa}{\kappa}\frac{\delta\Pi}{\Pi^*} - \frac{\delta\rho}{\rho^*} - \frac{\delta\Theta}{\Theta^*} &= -\left(\Pi^{\frac{1-\kappa}{\kappa}} - \frac{R}{P_0}\rho\Theta\right)_A^{n+1}.
\end{aligned}$$

Here reference profiles (Π^*, ρ^*, Θ^*) are taken to the be the domain average of the start of time step fields. Cubic Lagrange interpolation with a monotone limiter is used for the density and potential temperature fields and linear interpolation is used for the wind field. Again, 4 iterations of the iterative-semi-implicit method is used.

As a particular example we consider the *Sod shock tube test problem*. This is an example of a Riemann problem which describes the behaviour of a gas which is initially at two different densities and pressures, separated by a membrane. At time $t = 0$ this membrane is ruptured leading to a solution consisting of three waves: a shock moving into the region of low pressure; a rarefraction wave expanding into the high pressure region and a contact discontinuity between the two gas regions. The setup specifies

$$\begin{aligned}
p(x,0) &= \begin{cases} p_l & \text{for} \quad x < 0, \\ p_r & \text{for} \quad x \geq 0, \end{cases} \\
\rho(x,0) &= \begin{cases} \rho_l & \text{for} \quad x < 0, \\ \rho_r & \text{for} \quad x \geq 0, \end{cases} \\
u(x,0) &= u_0\,,
\end{aligned}$$

and the initial potential temperature is then diagnosed from the equation of state (18) with $R = 1$, $c_p = 2.4$ and $P_0 = 1$. The values used here follow [LeV02b]: $p_l \equiv \rho_l = 3$, $p_r \equiv \rho_r = 1$ and $u_0 = 0$.

This leads to a solution very similar to the dam break problem considered in the previous section but with the addition of the contact discontinuity that travels with characteristic speed u. The evolution of the contact discontinuity leads to a rightward moving interface, and this is appropriate for a moving mesh calculation. In particular the potential temperature Θ varies rapidly across this interface, and only sees the contact discontinuity. Accordingly by using a monitor function based only on Θ the moving mesh method can be made to ignore the fast shock and rarefraction waves and only adapt to the temperature change. This is a more relevant example for meteorological applications where it could be imagined that the mesh is adapting to some slowly evolving feature,

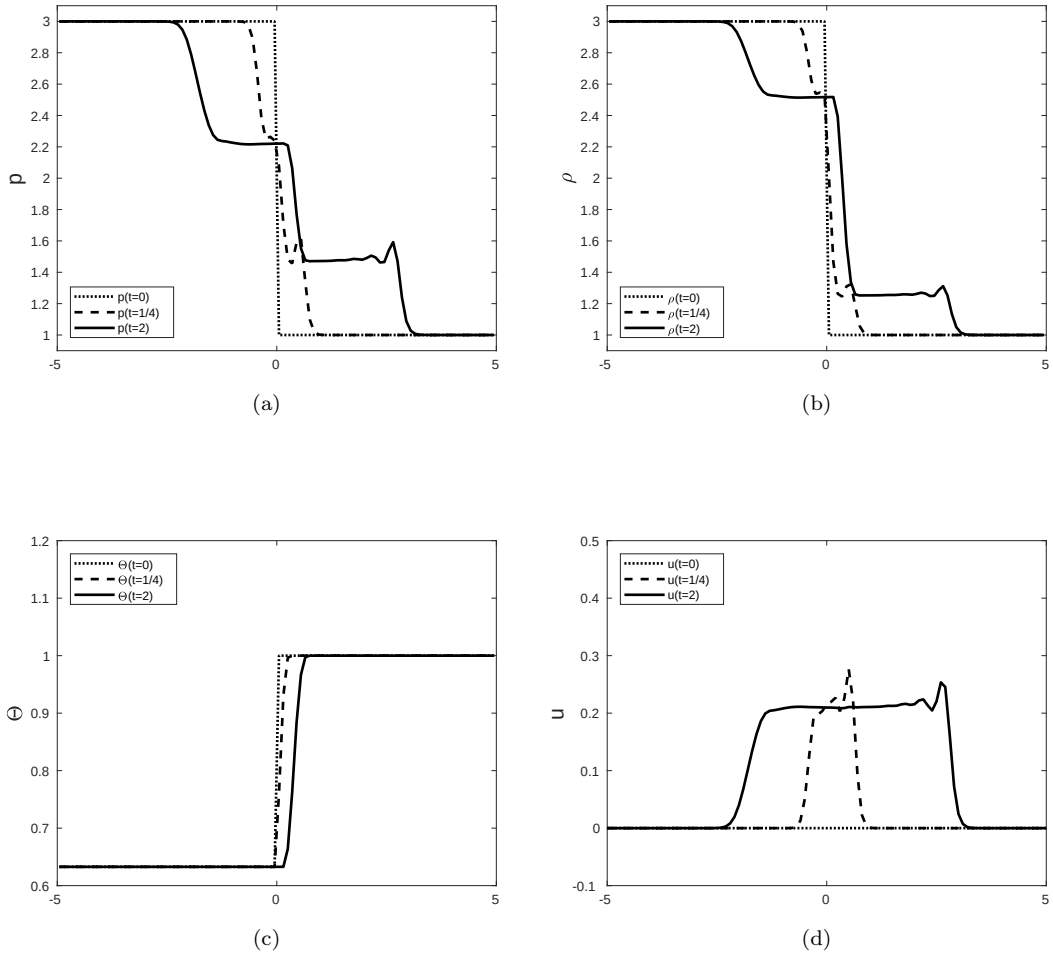


Figure 11: Evolution of the (a) pressure, (b) density, (c) potential temperature and (d) velocity field in the Sod shock tube test using a fixed spatial mesh.

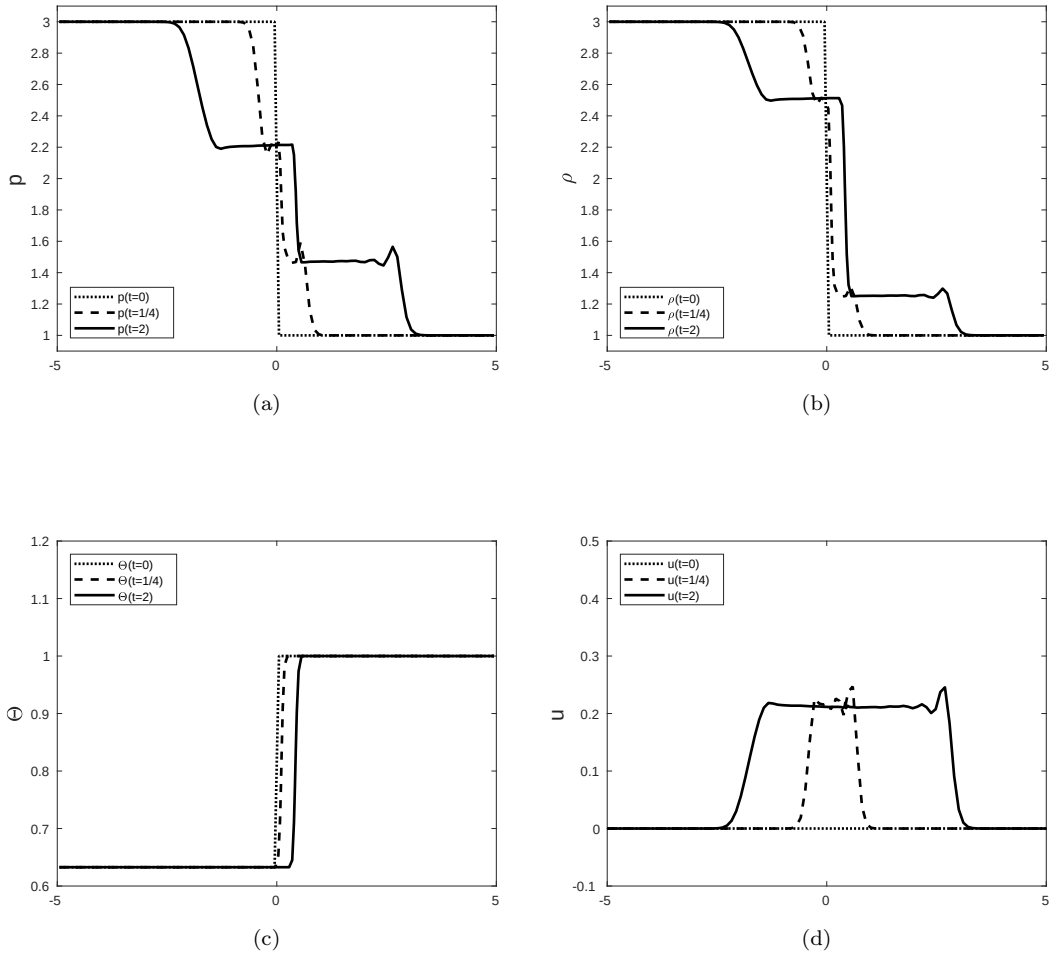


Figure 12: Evolution of the (a) pressure, (b) density, (c) potential temperature and (d) velocity field in the Sod shock tube test using a moving spatial mesh with an arc-length monitor function applied to the potential temperature field Θ.

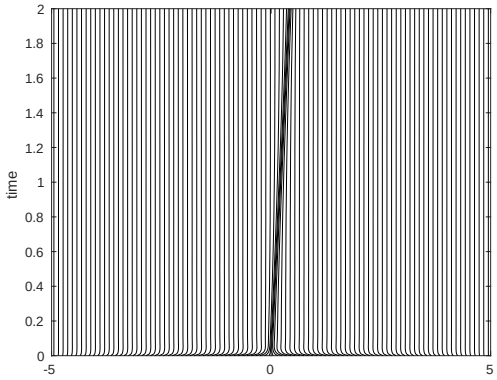


Figure 13: Evolution of the mesh for the Sod shock tube test presented in Figure 12.

that travels with the flow field, instead of the fast gravity and acoustic waves. Since, in equation (18) the potential temperature is purely advected by the flow, this is a similar application to Burgers' equation considered in other sections.

In Figures 11 and 12 we present the results of two calculations on a fixed and a moving mesh. In both cases we take $N_x = 100$ and $\Delta t = 10^{-2}$, and for the latter calculation we take a monitor function which is the arc-length of Θ. In Figure 13 we see the evolution of the moving mesh. We see that the static mesh has led to the contact discontinuity broadening in a similar manner to the front of the solution of Burgers' equation considered earlier. In contrast the adaptive mesh method has preserved a much sharper profile to the contact discontinuity. We also note that the leading edge of the solution is very similar in the two cases (with an oscillating front). This is because the use of the monitor function based on potential temperature has led to a mesh which is only refined close to the contact discontinuity and is otherwise uniform. To obtain a solution adapted to all three shocks the monitor function can be applied to the pressure field (along with a reduced time step, $\Delta t = 1/300$), yielding Figure 14 where now there is both an improvement in the shape of the contact discontinuity, compared to the fixed mesh, and also in the shock wave (reduced oscillations). The evolution of the mesh is shown in Figure 15 showing the adaption to the three distinct wave features.

5. Conclusions

In this paper we have implemented a location based moving mesh method **MM-SISL** which is a natural extension of the SISL method and involves little extra computational cost. The results from a series of tests of this method on problems

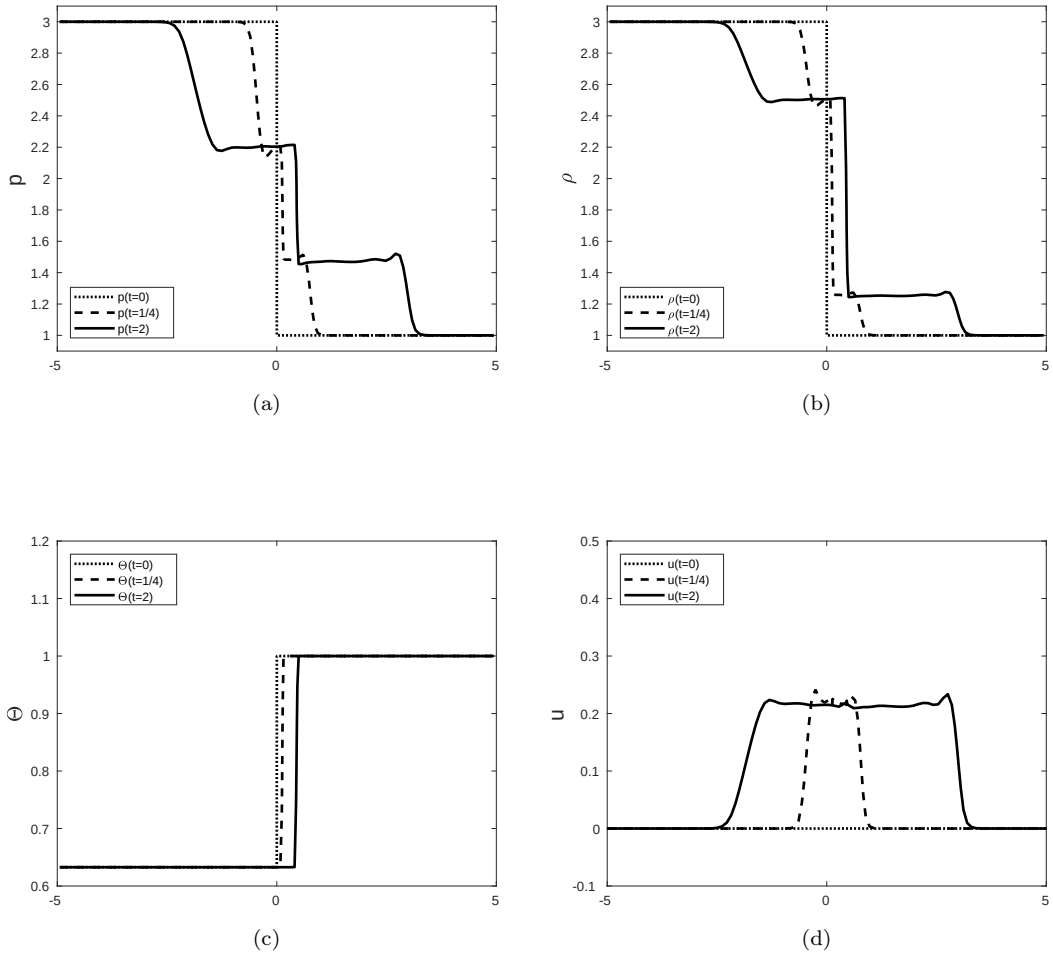


Figure 14: Evolution of the (a) pressure, (b) density, (c) potential temperature and (d) velocity field in the Sod shock tube test using a moving spatial mesh with an arc-length monitor function applied to the pressure field Θ.

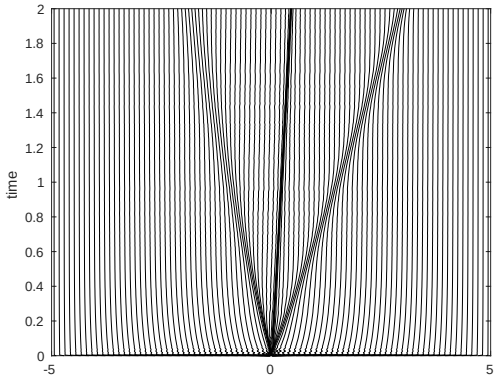


Figure 15: Evolution of the mesh for the Sod shock tube test presented in Figure 14.

in one-dimension this show a great improvement over the fixed mesh calculations, leading to low solution errors and stable solutions even when large time steps are taken. We conclude that the **MMSISL** method shows promise for future development although much work needs to be done on the development of suitable monitor functions and tuning the method to be stable in all cases.

In future work we will develop the **MMSISL** method for systems posed in more spatial dimensions, and will test it, in comparison with other methods (such as adaptive mesh refinement methods) on a much more challenging set of benchmark problems including problems in gas dynamics with shocks, and those which arise in a meteorological context.

Acknowledgement

This research was funded by an EPSRC CASE award with the Met Office. We are grateful to Prof. Nigel Wood of the Met Office for many useful discussions and to the anonymous referees for their very helpful comments.

Bibliography

[BCW13] C.J. Budd, M.J.P. Cullen, and E.J. Walsh, *Mongeampre based moving mesh methods for numerical weather prediction, with applications to the Eady problem*, J. Comp. Phys. **236** (2013), 247–270.

[BHR09] C.J. Budd, W. Huang, and R.D. Russell, *Adaptivity with moving grids*, Acta Numerica **18** (2009), no. 1, 111–241.

[CGM+98] J. Côté, S. Gravel, A. Méthot, A. Patoine, M. Roch, and A. Staniforth, *The operational CMC-MRB Global Environmental Multiscale (GEM) model. part I: Design considerations and formulation*, Monthly Weather Review **126** (1998), 1373–1395.

[CK09] E. Celledoni and B.K. Kometa, *Semi-Lagrangian Runge-Kutta exponential integrators for convection dominated problems*, Journal of Scientific Computing **41** (2009), no. 1, 139–164.

[CXY+08] D. Chen, J. Xue, X. Yang, H. Zhang, X. Shen, J. Hu, Y. Hu, L. Wang, J. LiRen, and J. Chen, *New generation of multi-scale NWP system (GRAPES): general scientific design*, Chinese Science Bulletin **7** (2008), 3433–3445.

[dBS77] C. de Boor and B. Swartz, *Piecewise monotone interpolation*, Journal of Approximation Theory **21** (1977), no. 4, 411–416.

[FC80] F.N. Fritsch and R.E. Carlson, *Monotone piecewise cubic interpolation*, SIAM Journal on Numerical Analysis **17** (1980), no. 2, 238–246.

[FL03] R. Fazio and R.J. LeVeque, *Moving-mesh methods for one-dimensional hyperbolic problems using CLAWPACK*, Computers and mathematics with applications **45** (2003), 273–298.

[GNQ14] Wei Guo, R.D. Nair, and Jing-Mei Qiu, *A conservative semi-lagrangian discontinuous galerkin scheme on the cubed sphere*, Monthly Weather Review **142** (2014), no. 1, 457–475.

[HEOC87] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes, III*, Journal of Computational Physics **71** (1987), no. 2, 231–303.

[HR10] W. Huang and R.D. Russell, *Adaptive moving mesh methods*, Applied Mathematical Sciences, Springer, 2010.

[Hym83] J.M. Hyman, *Accurate monotonicity preserving cubic interpolation*, SIAM Journal on Scientific and Statistical Computing **4** (1983), no. 4, 645–654.

[KSD12] C. Kühnlein, P.K. Smolarkiewicz, and A. Dörnbrack, *Modelling atmospheric flows with adaptive moving meshes*, Journal of Computational Physics **231** (2012), no. 7, 2741–2763.

[LeV02a] R.J. LeVeque, *Finite volume methods for hyperbolic problems*, vol. 31, Cambridge University Press, 2002.

[LeV02b] ———, *Finite volume methods for hyperbolic problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

[LOC94] Xu-Dong Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of computational physics **115** (1994), no. 1, 200–212.

[McC88] J.D. McCalpin, *A quantitative analysis of the dissipation inherent in semi-Lagrangian advection*, Monthly Weather Review **116** (1988), no. 11, 2330–2336.

[McD84] A. McDonald, *Accuracy of multiply-upstream, semi-Lagrangian advective schemes*, Monthly Weather Review **112** (1984), no. 6, 1267–1275.

[McD87] ———, *Accuracy of multiply-upstream semi-Lagrangian advective schemes II*, Monthly Weather Review **115** (1987), no. 7, 1446–1450.

[MM94] K.W. Morton and D.F. Mayers, *Numerical solutions of partial differential equations: An introduction*, Cambridge University Press, 1994.

[PC11] C. Piccolo and M.J.P. Cullen, *Adaptive mesh method in the met office variational data assimilation system*, Quarterly Journal of the Royal Meteorological Society **137** (2011), no. 656, 631–640.

[Rob81] A. Robert, *A stable numerical integration scheme for the primitive meteorological equations*, Atmos.-Ocean **19** (1981), 35–46.

[Rob82] ———, *A semi-Lagrangian and semi-implicit numerical integration scheme for the primitive meteorological equations*, Japan Meteor. Soc. **60** (1982), 319–325.

[RTS+95] H. Ritchie, C. Temperton, A. Simmons, M. Hortal, T. Davies, D. Dent, and M. Hamrud, *Implementation of the semi-Lagrangian method in a high resolution version of the ECMWF forecast model*, Monthly Weather Review **123** (1995), 489–514.

[RW90] P.J. Rasch and D.L. Williamson, *On shape-preserving interpolation and semi-Lagrangian transport*, SIAM Journal on Scientific and Statistical Computing **11** (1990), no. 4, 656–687.

[SC91] A. Staniforth and J. Côté, *Semi-lagrangian integration schemes for atmospheric models - a review*, Monthly Weather Review **119** (1991), no. 9, 2206–2223.

[Smi00] C.J. Smith, *The semi-lagrangian method in atmospheric modelling*, Ph.D. thesis, University of Reading, 2000.

[THA01] C. Temperton, M. Hortal, and Simmons A.J., *A two-time-level semi-Lagrangian global spectral model*, Quarterly Journal of the Royal Meteorological Society **127** (2001), 111–126.

[TT03] Huazhong Tang and Tao Tang, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM Journal on Numerical Analysis **41** (2003), no. 2, 487–515.

[TX07] Tao Tang and Jinchao Xu (eds.), *Adaptive computations: Theory and algorithms*, Mathematics Monograph Series 6, Science Press, 2007.

[WSW+14] N. Wood, A. Staniforth, A. White, T. Allen, M. Diamantakis, M. Gross, T. Melvin, C. Smith, S. Vosper, M. Zerroukat, et al., *An inherently mass-conserving semi-implicit semi-lagrangian discretization of the deep-atmosphere global non-hydrostatic equations*, Quarterly Journal of the Royal Meteorological Society **140** (2014), no. 682, 1505–1520.

[Zeg07] P.A. Zegeling, *Theory and application of adaptive moving grid methods*, in Tang and Xu [TX07], Chapter 7, pp. 279–332.